Package 'SmarterPoland'

January 20, 2025

Type Package

Title Tools for Accessing Various Datasets Developed by the Foundation SmarterPoland.pl

Version 1.8.1

Author Przemyslaw Biecek

Maintainer Przemyslaw Biecek <przemyslaw.biecek@gmail.com>

Description Tools for accessing and processing datasets prepared by the Foundation SmarterPoland.pl. Among all: access to API of Google Maps, Central Statistical Office of Poland, MojePanstwo, Eurostat, WHO and other sources.

LazyLoad yes

LazyData yes

LazyDataCompression xz

License GPL-3

Depends R (>= 3.0), ggplot2, httr, htmltools

Imports jsonlite, rjson

NeedsCompilation no

Repository CRAN

Date/Publication 2023-08-20 21:22:35 UTC

Contents

SmarterPoland-package 2
Bank Danych Lokalnych 3
cities_lon_lat 4
countries 5
Dark Sky API for weather forecast 5
getEurostatDictionary 7
getEurostatRaw 8
getEurostatRCV 9
getEurostatTOC 10
getGoogleMapsAddress 11

getMillwardBrown 12
grepEurostatTOC 12
maturaExam 13

Index **14**

`SmarterPoland-package` *Tools for Accessing Various Datasets Developed by the Foundation SmarterPoland.pl*

Description

Tools for accessing and processing datasets prepared by the Foundation SmarterPoland.pl. Among all: access to API of Google Maps, Central Statistical Office of Poland, Eurostat, WHO and other sources.

Author(s)

Author: Przemyslaw Biecek Maintainer: Przemyslaw Biecek <przemyslaw.biecek@gmail.com>

See Also

`getMillwardBrown`, `getEurostatRCV`, `getBDLseries`, `getWeatherForecast`

Examples

```
## Not run:
 # download the dataset 'Pupil/Student - teacher ratio and average class' from eurostat
 # for more developed API see https://github.com/rOpenGov/eurostat
 tmp <- getEurostatRCV(kod = "educ_iste")
 head(tmp)

 # download the dataset 'People killed in road accidents' from eurostat
 # and plot a maptable for selected countries
 # for more developed API see https://github.com/rOpenGov/eurostat
 library(ggplot2)
 t1 <- getEurostatRCV("tsdtr420")
 t1 <- t1[t1$geo
 ggplot(t1, aes(time, value, color=sex, group=sex)) +
  geom_line() + facet_wrap(~geo)

## End(Not run)
```

`Bank Danych Lokalnych` *API to Bank Danych Lokalnych [GUS]*

Description

Access to the GUS Bank Danych Lokalnych with the use of API developed by MojePanstwo.

Download and parse data from Bank Danych Lokalnych with the use of API developed by MojePanstwo.

Usage

```
getBDLtree(raw = FALSE, debug = 0)
getBDLsearch(query = "", debug = 0, raw = FALSE)
getBDLseries(metric_id = "", slice = NULL, time_range = NULL,
            wojewodztwo_id = NULL, powiat_id = NULL, gmina_id = NULL,
            meta = NULL, debug = 0, raw = FALSE)
getMPgminy(debug = 0)
getMPpowiaty(debug = 0)
getMPwojewodztwa(debug = 0)
```

Arguments

`debug`	Level of debug info. 0 for no debug, 1 or 2 for info about processed groups.
`raw`	If raw = TRUE the resulting JSON is returned without any transformation. For raw = FALSE results are transformed into a data.frame.
`query`	A query for DBL search.
`metric_id`	Metric id, if unknown then look for it in DBL tree or DBL search.
`slice`	A table with id dimensions, with format [1,34,*]. Use '*' to choose all dimensions (or use an empty string).
`time_range`	Year or range (like 2000:2010), empty means - full range.
`wojewodztwo_id`	Voievodship id or '*' for all.
`powiat_id`	County id of '*' for all. It's internal ID. Use `getMPpowiaty()` to get names and other information.
`gmina_id`	Subcounty id or '*' for all. It's internal ID. Use `getMPgminy()` to get TERYT codes.
`meta`	Should meta data be returned?

Value

The function `getMPgminy()` returns a data frame with identifiers id/TERYT for each subcounty. The function `getMPpowiaty()` returns a data frame with identifiers id for each county.

The function `getBDLtree()` returns a data frame with identifiers of resources in Bank Danych Lokalnych.

Author(s)

Przemyslaw Biecek

References

The API of Bank Danych Lokalnych developed by MojePanstwo is described as https://mojepanstwo.pl/api/dane/get_

Examples

```
## Not run:
 # the data is downloaded and parsed from Internet
 # not that this dataset is pre-calculated in the package
 BDLtree <- getBDLtree(2)
 head(BDLtree)

 DBLtransport <- getBDLsearch("transport")
 head(DBLtransport)

 BDLseries <- getBDLseries(metric_id = 1)
 head(BDLseries)

 gminy <- getMPgminy()
 head(gminy)

 powiaty <- getMPpowiaty()
 head(powiaty)

## End(Not run)
```

---

cities_lon_lat *Geocoordinates of Largest Cities*

Description

A subset of world.citiesmaps. Extracted in order to shink number of dependencies. Only cities with pop > 50k are keept.

Author(s)

Przemyslaw Biecek [based on world.cities]

Examples

```
## Not run:
library(maps)
data(world.cities)
cities_lon_lat <- world.cities[!duplicated(world.cities$name),]
rownames(cities_lon_lat) = cities_lon_lat[,1]
cities_lon_lat <- cities_lon_lat[cities_lon_lat$pop > 50000,]
```

```
cities_lon_lat <- cities_lon_lat[,4:5]

## End(Not run)
```

`countries` *Birth and death rates, continent and population for selected countries*

Description

Data from World Health Organization database http://apps.who.int/gho/data/view.main.CBDR2040. Based on the example from Grammar of Graphics by Leland Wilkinson.

Author(s)

Przemyslaw Biecek [based on WHO data]

Examples

```
## Not run:
library(maps)
data(countries)
head(countries)

## End(Not run)
```

`Dark Sky API for weather forecast`
Access to Weather Forecasts with the Use of Dark Sky API.

Description

Access to hourly and daily weather forecasts with the use of Dark Sky API.

Usage

```
getWeatherForecast(apiKey, lat = NA, lon = NA, city = NA, raw=FALSE)
```

Arguments

`apiKey`	You need to have Dark Sky apiKey in order to access weather forecasts. See here: https://developer.forecast.io/ hor more details.
`lat`	The latitude coordinate for which prediction has to be made.
`lon`	The longitude coordinate for which prediction has to be made.
`city`	Instead of lat and lon you may specify name of the city for which prediction has to be made.
`raw`	If TRUE then no parsing is done. The function getWeatherForecast() just download an forecast and returns it as a list.

Value

The function `getWeatherForecast()` returns list of three datasets. now and by.hour datasets contains predictions. For each timepoint following information are collected:

time, summary, icon, precipIntensity, precipProbability, temperature, apparentTemperature, dewPoint, humidity, windSpeed, windBearing, visibility, cloudCover, pressure, ozone, temperatureCelsius, apparentTemperatureCelsius

Daily predictions (by.day component) contain following information:

time, summary, icon, sunriseTime, sunsetTime, moonPhase, precipIntensity, precipIntensityMax, precipProbability, temperatureMin, temperatureMinTime, temperatureMax, temperatureMaxTime, apparentTemperatureMin, apparentTemperatureMinTime, apparentTemperatureMax, apparentTemperatureMaxTime, dewPoint, humidity, windSpeed, windBearing, visibility, cloudCover, pressure, ozone, precipIntensityMaxTime, precipType, temperatureMaxCelsius, temperatureMinCelsius, apparentTemperatureMaxCelsius, apparentTemperatureMinCelsius

Author(s)

Przemyslaw Biecek

References

The Dark Sky API for weather forecasts is described as `https://developer.forecast.io/`

Examples

```
## Not run:
 # you have to have apiKey to execute these examples
library(scales)
library(ggplot2)

prognoza <- getWeatherForecast(apiKey, city='Warsaw')

ggplot(prognoza$by.hour, aes(y=temperatureCelsius, x=time)) +
  geom_line() + geom_point() +
  geom_point(data=prognoza$now, size=10, color='red') +
  theme(title=element_text(size=20),
        axis.text=element_text(size=20)) +
  scale_x_datetime(breaks = date_breaks("3 hour"),
                   minor_breaks = date_breaks("1 hour"),
                   labels = date_format("
  ylab("") + xlab("") + ggtitle("Prognoza temperatury dla Warszawy")


## End(Not run)
```

`getEurostatDictionary` *Download a Dictionary from the Eurostat Database*

Description

Download a dictionary for given coded variable from Eurostat (ec.europa.eu/eurostat).

Usage

```
getEurostatDictionary(dictname)
```

Arguments

`dictname`	Character, dictionary for given variable name will be downloaded.

Value

A data.frame with two columns, first with code names and second with full names.

Author(s)

Przemyslaw Biecek

References

The TOC is downloaded from the `http://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadList`

See Also

See Also as `getEurostatRCV`, `getEurostatRaw`, `grepEurostatTOC`.

Examples

```
## Not run:
 tmp <- getEurostatDictionary("crop_pro")
 head(tmp)

## End(Not run)
```

getEurostatRaw *Download a Dataset from the Eurostat Database*

Description

Download a dataset from the eurostat database. The dataset is transformed into the tabular format.

Usage

```
getEurostatRaw(kod = "educ_iste", rowRegExp=NULL, colRegExp=NULL,
       strip.white = TRUE)
```

Arguments

kod	A code name for the data set of interested. See the table of contents of eurostat datasets for more details.
rowRegExp	If not NULL this regular expression will be used to filter rows out of downloaded file.
colRegExp	If not NULL this regular expression will be used to filter collumns out of downloaded file.
strip.white	Passed to the internal `read.table()`. By default it strips white spaces from eurostat values.

Value

A dataset in data.frame format. First column contains names of cases. Column names usually corresponds to years.

Author(s)

Przemyslaw Biecek

References

Data is downloaded from `http://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadListing` website.

See Also

See Also as `getEurostatTOC`, `getEurostatRaw`, `grepEurostatTOC`.

Examples

```
## Not run:
 tmp <- getEurostatRaw(kod = "educ_iste")
 head(tmp)

## End(Not run)
```

getEurostatRCV *Download a Dataset from the Eurostat Database*

Description

Download a dataset from the eurostat database. The dataset is transformed into the molten / row-column-value format (RCV).

Usage

```
getEurostatRCV(kod = "educ_iste", ...)
```

Arguments

kod	A code name for the data set of interested. See the table of contents of eurostat datasets for more details.
...	Other parameters that are passed to getEurostatRaw().

Value

A dataset in the molten format with the last column 'value'.

Author(s)

Przemyslaw Biecek

References

Data is downloaded from http://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadListing website.

See Also

See Also as getEurostatTOC, getEurostatRaw, grepEurostatTOC.

Examples

```
## Not run:
 tmp <- getEurostatRCV(kod = "educ_iste")
 head(tmp)

 t1 <- getEurostatRCV("rail_ac_catvict")
 tmp <- cast(t1, geo ~ time , mean, subset=victim=="KIL" &
                pers_inv=="TOTAL" & accident=="TOTAL")
 tmp3 <- tmp[,1:9]
 rownames(tmp3) <- tmp3[,1]
 tmp3 <- tmp3[c("UK", "SK", "FR", "PL", "ES", "PT", "LV"),]
 matplot(2005:2012,t(tmp3[,-1]), type="o", pch=19, lty=1,
                las=1, xlab="", ylab="", yaxt="n")
```

```
 axis(2,tmp3[,9], rownames(tmp3), las=1)

## End(Not run)
```

---

getEurostatTOC *Eurostat Table of Contents*

Description

Download a table of contents of eurostat datasets. Note that the values in column 'code' should be used to download a selected dataset.

Usage

```
getEurostatTOC()
```

Value

A data.frame with eight columns

title	The name of dataset of theme
code	The codename of dataset of theme, will be used by the getEurostatRCV and getEurostatRaw functions.
type	Is it a dataset, folder or table.
last.update.of.data, last.table.structure.change, data.start, data.end	Dates.

Author(s)

Przemyslaw Biecek

References

The TOC is downloaded from the http://ec.europa.eu/eurostat/estat-navtree-portlet-prod/BulkDownloadLis

See Also

See Also as getEurostatRCV, getEurostatRaw, grepEurostatTOC.

Examples

```
## Not run:
 tmp <- getEurostatTOC()
 head(tmp)

## End(Not run)
```

`getGoogleMapsAddress` *Geolocalisation with Google Maps*

Description

Get geolocalisation (longitude, latitude) of a given adress with the use of Google Maps API.

The Google Maps API is used to determine the geolocalisation (longitude, latitude) of a given adress.

Usage

```
getGoogleMapsAddress(street = "Banacha 2", city = "Warszawa",
country="Poland", positionOnly = TRUE, delay=1)
```

Arguments

`street`	An address (street and building number)
`city`	City
`country`	Country
`positionOnly`	What should be returned, vector with longitude, latitude coordinates or the raw result from Google Maps API
`delay`	Number of seconds to wait between api calls

Value

If positionOnly=TRUE then a vector with two values or a raw list from Google Maps otherwise.

Author(s)

Przemyslaw Biecek

References

The Google Maps API `https://developers.google.com/maps/`

Examples

```
## Not run:
 getGoogleMapsAddress()

## End(Not run)
```

getMillwardBrown *MillwardBrown Pool Results*

Description

Download pool results from MillwardBrown website.

Usage

```
getMillwardBrown()
```

Value

A dataset in the molten format with pool date, party and percent of votes.

Author(s)

Maciej Beresewicz [data extraction] Przemyslaw Biecek [data melting]

Examples

```
## Not run:
 getMillwardBrown()

## End(Not run)
```

grepEurostatTOC *Names of Eurostat Datasets That Fit Given Pattern*

Description

Lists names of dataset from eurostat with the particular pattern in the description.

This function downloads list of all datasets available on eurostat and return list of names of datasets that contains particular pattern in the dataset description.

E.g. all datasets related to education of teaching.

Usage

```
grepEurostatTOC(pattern)
```

Arguments

pattern	Character, only datasets that contains this pattern in the description will be returned.

Value

A data.frame with eight columns

`title`	The name of dataset of theme
`code`	The codename of dataset of theme, will be used by the getEurostatRCV and getEurostatRaw functions.
`type`	Is it a dataset, folder or table.
`last.update.of.data`, `last.table.structure.change`, `data.start`, `data.end`	Dates.

Author(s)

Przemyslaw Biecek

See Also

See Also as `getEurostatRCV`, `getEurostatRaw`, `getEurostatTOC`.

Examples

```
## Not run:
 tmp <- grepEurostatTOC("education")
 head(tmp)

## End(Not run)
```

---

maturaExam *Results from Matura Exams in Poland for Math and Language for Years 2010-2015*

Description

This dataset is created based on data from ZPD package, see `https://github.com/zozlak/ZPD` and `http://zpd.ibe.edu.pl/doku.php?id=obazie`. Each row shows results for one person that takes matura exams in a given year.

Author(s)

Przemyslaw Biecek [based on IBE / ZPD data]

Examples

```
## Not run:
data(maturaExam)
head(maturaExam)

## End(Not run)
```

Index

∗ **database**
Bank Danych Lokalnych, 3
Dark Sky API for weather forecast, 5
getEurostatDictionary, 7
getEurostatRaw, 8
getEurostatRCV, 9
getEurostatTOC, 10
grepEurostatTOC, 12
∗ **datasets**
cities_lon_lat, 4
countries, 5
maturaExam, 13
∗ **geolocalisation**
getGoogleMapsAddress, 11
∗ **package**
SmarterPoland-package, 2

Bank Danych Lokalnych, 3
BDLtree *(*Bank Danych Lokalnych*)*, 3

cities_lon_lat, 4
countries, 5

Dark Sky API for weather forecast, 5

getBDLsearch *(*Bank Danych Lokalnych*)*, 3
getBDLseries, *2*
getBDLseries *(*Bank Danych Lokalnych*)*, 3
getBDLtree *(*Bank Danych Lokalnych*)*, 3
getEurostatDictionary, 7
getEurostatRaw, *7, 8*, 8, *9, 10*, *13*
getEurostatRCV, *2*, *7*, 9, *10*, *13*
getEurostatTOC, *8, 9*, 10, *13*
getGoogleMapsAddress, 11
getMillwardBrown, *2*, 12
getMPgminy *(*Bank Danych Lokalnych*)*, 3
getMPpowiaty *(*Bank Danych Lokalnych*)*, 3
getMPwojewodztwa *(*Bank Danych Lokalnych*)*, 3
getWeatherForecast, *2*
getWeatherForecast *(*Dark Sky API for weather forecast*)*, 5
grepEurostatTOC, *7–10*, 12

maturaExam, 13

SmarterPoland *(*SmarterPoland-package*)*, 2
SmarterPoland-package, 2